

**AMENDMENTS TO THE CLAIMS**

The listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. 1. (Currently Amended) A computer implemented method of processing an application that includes a plurality of application execution threads for grouping ~~processors~~ in a computer environment that includes a plurality of dissimilar processors, said method comprising:
 

receiving a resource request from ~~the~~ an application that is running a first application execution thread on a first processor type;

assigning one or more second processor types and a memory space to a group in response to the resource request, wherein the first processor type shares the memory space with the assigned second processor types, and wherein the first processor and the assigned second processor types are heterogeneous; and

executing processing on a second application execution thread using on at least one of the second processor types assigned to the group, the application execution thread running on the first processor type and corresponding to the application.
2. (Canceled)
3. (Original) The method as described in claim 1 further comprising:
 

identifying whether the application requests the memory space to be a private memory, wherein the private memory is accessible only by the assigned second processor types; and

classifying the memory space as the private memory.

4. (Original) The method as described in claim 3 further comprising:  
retrieving data from the private memory using one of the assigned second processor types;  
manipulating the data using one of the assigned second processor types, the manipulating resulting in resultant data; and  
storing the resultant data in a shared memory, the shared memory accessible by the first processor type.
5. (Original) The method as described in claim 1 further comprising:  
retrieving an affinity selection bit from the application;  
determining whether the application requests affinity processor selection based upon the affinity selection bit; and  
performing the assigning using affinity processor selection.
6. (Original) The method as described in claim 5 wherein the performing further comprises:  
selecting one of the second processor types based upon the affinity processor selection;  
determining whether the selected second processor type is available; and  
performing the assigning based upon the selected second processor type's availability.
7. (Original) The method as described in claim 1 further comprising:  
detecting that one or more of the second processor types are in use by an active execution thread;  
identifying an active priority that corresponds to the active execution thread;

comparing the active priority to a requesting priority, the requesting priority corresponding to the application execution thread; and

terminating the active execution thread if the active priority is lower than the requesting priority.

8. (Original) The method as described in claim 1 wherein the group corresponds to one or more group properties, wherein the group properties are selected from the group consisting of a sharing mode, a priority, and a scheduling policy.
9. (Original) The method as described in claim 1 wherein the group includes a plurality of second processors.
10. (Original) The method as described in claim 1 wherein the first processor type is a processing unit and wherein the second processor types are synergistic processing units.
11. (Currently Amended) An information handling system comprising:
  - a plurality of dissimilar processors;
  - a memory accessible by the plurality of dissimilar processors;
  - one or more nonvolatile storage devices accessible by the plurality of dissimilar processors; and
  - a ~~processor-grouping tool for processing an application that includes a plurality of application execution threads compiling source-code~~, the processor-grouping tool comprising software code effective to:
    - receive a resource request from the an application that is running a first application execution thread on a first processor type, the first processor type included in the plurality of dissimilar processors;

assign one or more second processor types included in the plurality of dissimilar processor types and a memory space included in the memory to a group in response to the resource request, wherein the first processor type shares the memory space with the assigned second processor types, and wherein the first processor and the assigned second processor types are heterogeneous; and

execute process an a second application execution thread using on at least one of the second processor types assigned to the group, the application execution thread running on the first processor type and corresponding to the application.

12. (Canceled)
13. (Original) The information handling system as described in claim 11 wherein the software code is further effective to:  
  
 identify whether the application requests the memory space to be a private memory, wherein the private memory is accessible only by the assigned second processor types; and  
  
 classify the memory space as the private memory.
14. (Original) The information handling system as described in claim 13 wherein the software code is further effective to:  
  
 retrieve data from the private memory using one of the assigned second processor types;  
  
 manipulate the data using one of the assigned second processor types whereby the data manipulation results in resultant data; and  
  
 store the resultant data in a shared memory included in the memory, the shared memory accessible by the first processor type.

15. (Original) The information handling system as described in claim 11 wherein the software code is further effective to:  
  
retrieve an affinity selection bit from the application;  
  
determine whether the application requests affinity processor selection based upon the affinity selection bit; and  
  
perform the assigning using affinity processor selection.
16. (Original) The information handling system as described in claim 15 wherein the software code is further effective to:  
  
select one of the second processor types based upon the affinity processor selection;  
  
determine whether the selected second processor type is available; and  
  
perform the assigning based upon the selected second processor type's availability.
17. (Original) The information handling system as described in claim 11 wherein the software code is further effective to:  
  
detect that one or more of the second processor types are in use by an active execution thread;  
  
identify an active priority that corresponds to the active execution thread;  
  
compare the active priority to a requesting priority, the requesting priority corresponding to the application execution thread; and  
  
terminate the active execution thread if the active priority is lower than the requesting priority.
18. (Original) The information handling system as described in claim 11 wherein the group corresponds to one or more group properties, wherein the group properties

are selected from the group consisting of a sharing mode, a priority, and a scheduling policy.

19. (Original) The information handling system as described in claim 11 wherein the group includes a plurality of second processors.
20. (Original) The information handling system as described in claim 11 wherein the first processor type is a processing unit and wherein the second processor types are synergistic processing units.
21. (Currently Amended) A computer program product stored on a computer ~~readable~~ operable media ~~for grouping processors for a~~, the computer program product including a set of instructions that, when executed by an information handling system, causes the information handling system to perform steps said computer program product comprising:  
~~means for~~ receiving a resource request from an application that is running a first application execution thread on a first processor type;  
~~means for~~ assigning one or more second processor types and a memory space to a group in response to the resource request, wherein the first processor type shares the memory space with the assigned second processor types, and wherein the first processor and the assigned second processor types are heterogeneous; and  
~~means for~~ processing an executing a second application execution thread using on at least one of the second processor types assigned to the group, ~~the application execution thread running on the first processor type and corresponding to the application.~~
22. (Canceled)

23. (currently amended) The computer program product as described in claim 21 wherein the set of instructions, when executed by the information handling system, cause the information handling system to perform further steps comprising:
- ~~means for~~ identifying whether the application requests the memory space to be a private memory, wherein the private memory is accessible only by the assigned second processor types; and
- ~~means for~~ classifying the memory space as the private memory.
24. (currently amended) The computer program product as described in claim 23 wherein the set of instructions, when executed by the information handling system, cause the information handling system to perform further steps comprising:
- ~~means for~~ retrieving data from the private memory using one of the assigned second processor types;
- ~~means for~~ manipulating the data using one of the assigned second processor types, the manipulating resulting in resultant data; and
- ~~means for~~ storing the resultant data in a shared memory, the shared memory accessible by the first processor type.
25. (currently amended) The computer program product as described in claim 21 wherein the set of instructions, when executed by the information handling system, cause the information handling system to perform further steps comprising:
- ~~means for~~ retrieving an affinity selection bit from the application;
- ~~means for~~ determining whether the application requests affinity processor selection based upon the affinity selection bit; and
- ~~means for~~ performing the assigning using affinity processor selection.

26. (currently amended) The computer program product as described in claim 25 wherein the performing further includes steps comprises comprising:  
  
means for selecting one of the second processor types based upon the affinity processor selection;  
  
means for determining whether the selected second processor type is available;  
and  
  
means for performing the assigning based upon the selected second processor type's availability.
27. (currently amended) The computer program product as described in claim 21 wherein the set of instructions, when executed by the information handling system, cause the information handling system to perform further steps comprising:  
  
means for detecting that one or more of the second processor types are in use by an active execution thread;  
  
means for identifying an active priority that corresponds to the active execution thread;  
  
means for comparing the active priority to a requesting priority, the requesting priority corresponding to the application execution thread; and  
  
means for terminating the active execution thread if the active priority is lower than the requesting priority.
28. (Original) The computer program product as described in claim 21 wherein the group corresponds to one or more group properties, wherein the group properties are selected from the group consisting of a sharing mode, a priority, and a scheduling policy.



29. (Original) The computer program product as described in claim 21 wherein the group includes a plurality of second processors.
30. (Original) The computer program product as described in claim 21 wherein the first processor type is a processing unit and wherein the second processor types are synergistic processing units.